УДК 004.93

Д.Ю. Коваль, В.М. Синєглазов, О.І. Чумаченко
Національний Авіаційний Університет, Україна
пр. Космонавта Комарова, 1, м. Київ, 03680

СИНТЕЗ ОПТИМАЛЬНОЇ СТРУКТУРИ ГЛИБОКОЇ НЕЙРОННОЇ МЕРЕЖІ

D.Y. Koval, V.M. Sineglazov, O.I. Chumachenko
National Aviation University, Ukraine
Kosmonavta Komarova av., 1, Kiev, 03680

SYNTHESIS OF OPTIMAL STRUCTURE OF DEEP NEURAL NETWORK

Запропоновано використання еволюційних алгоритмів для формування топології глибоких нейронних мереж.
Ключові слова: Парето, глибокі нейронні мережі, генетичний алгоритм.
The use of evolutionary algorithms for synthesis of topologies of deep neural networks is proposed.
Keywords: Pareto, deep neural networks, genetic algorithm.

Вступ

Штучні нейронні мережі – це структури, сформовані об’єднанням нейронів, які організовані у вхідний, вихідний та приховані шари. Нейронні мережі – потужний інструмент, який може бути використаний для вирішення широкого спектру задач.

В даний час широке розповсюдження знайшли глибокі нейронні мережі, які складаються з двох частин: групи автоасоціативних мереж (автоенкодерів або машин Больцмана), які виконують попереднє навчання основної мережі, і основної мережі (багатошарового перцептрону), яка займається розв’язанням поставленої задачі. Однак, синтез оптимальної структури такої нейронної мережі (визначення кількості шарів, нейронів, зв’язків між нейронами) для вирішення поставленої задачі є доволі непростим завданням, оскільки, з одного боку, необхідно забезпечити якомога більшу точність вирішення поставленої задачі, а одним з факторів, які впливають на точність роботи є саме кількість нейронів. З іншого боку, для уникнення проблеми перенавчання та зменшення обчислювальної складності та підвищення швидкості роботи, необхідно мінімізувати кількість нейронів та зв’язків між ними. Зазвичай подібна дилема вирішується дизайнером нейромережі шляхом перебору різних конфігурацій ШНМ.

В даній роботі пропонується для синтезу оптимальної структури штучних нейронних мереж скористатися еволюційними алгоритмами розв’язання задач багатокритеріальної оптимізації.

Постановка задачі

Визначити структуру глибокої нейронної мережі на основі навчальної вибірки, яка складається з пар (x_i, d_i), де x_i — вхідне значення, d_i — бажаний вихід мережі.

Критерієм якості вирішення поставленої задачі виступають два критерії:
1) мінімізація середньої квадратичної помилки роботи мережі. Обчислюється на основі перевірочної вибірки.

$$I_1 = \sum_{i=1}^{n} (y_i - d_i)^2$$

де y_i — отриманий вихід нейромережі, n — кількість елементів вибірки.
2) мінімізація загальної кількості нейронів у всіх шарах

$$I_2 = \sum_{j=1}^{k} N_j$$

де k — кількість шарів в нейромережі, N_j — кількість нейронів у шарі j.

Окрім того на розв'язок даної задачі накладаються обмеження:

1) Максимальна кількість шарів K_{max}
2) Максимальна кількість нейронів у кожному з шарів N_{max}
3) Максимальне допустиме значення помилки ε_{max}

Схема гібридного алгоритму для вирішення умовних багатокритеріальних задач оптимізації

Для вирішення поставленої задачі умовної багатокритеріальної оптимізації структури нейронної мережі пропонується використати гібридний еволюційний алгоритм, опис якого представлено нижче.

Вхід: N (розмір популяції), N_A (розмір архіву), T (максимальне число поколінь), p_c (ймовірність схрещування), p_m (ймовірність мутації).

Вихід: A (множина недомінованих особин).

Крок 0. Ініціалізація: Випадковим чином генерується початкова популяція P_0. Хромосоми, що кодують кожну можливу топологію нейронної мережі мають вигляд:

$$W_i(w_{11} \ldots w_{1k}\ w_{21} \ldots\ w_{2k}\ \ldots\ w_{n1} \ldots\ w_{nk}),$$

де $w_{p1} \ldots w_{pk}$ - закодований у двійковому форматі вектор, що описує кількість нейронів у p-тому шарі нейронної мережі, n — кількість шарів, k — кількість біт, що використовується для кодування кожного значення (гена).

Кожна хромосома буде відповідати певній архітектурі нейромережі. Кількість нейронів у першому та останньому шарі може визначатися задачею. В такому разі випадковим чином генеруються тільки решта компонент кожного вектора.

Також створюємо порожній архів $A_0 = \varnothing$ і задаємо $t = 0$.

Крок 1. Визначення пристосованості: Для кожної хромосоми з популяції та архіву будуємо відповідну їй нейронну мережу. Тобто, хромосомі $W_i(w_1\ w_2\ \ldots\ w_n)$ відповідатиме нейронна мережа, в якій буде w_1 нейронів у першому шарі, … w_p нейронів у p-тому шарі і тд, $p=1..n$.

Навчаємо кожну з отриманих мереж будь-яким з допустимих методів навчання. Для кожної з нейромереж обчислюємо середню квадратичну помилку E_p її роботи на контрольній вибірці даних та суму нейронів у всіх шарах S_n

Обчислена для кожної нейромережі помилка та сума нейронів використовується для визначення ефективного по Парето рішення та знаходження значень функцій пристосованості для кожної хромосоми відповідно до описаної нижче процедури:

1.1. Кожній особині в архіві A_t та популяції P_t присвоюється значення сили $S(i)$, яке представляє кількість особин, які дана особина домінує:

$$S(i) = \left|\{ j \mid j \in P_t \cup A_t \wedge i \succ j \}\right|$$

де операція |.| визначає кількість елементів у множині, $i = \overline{1,N},\ j = \overline{1,N}$

1.2. На основі значення $S(i)$ розраховується "грубе" (raw) значення функції пристосованості $R(i)$ особини i:

$$R(i) = \sum_{j \in P_t \cup A_t,\ j \succ i} S(j)$$

1.3. *Для кожної особини i* вираховується декартова відстань від неї до решти

особин *j* в архіві та популяції. Всі розраховані відстані для даної особини додаються у список і сортуються за зростанням. *k*-тий елемент такого списку для особини *i* позначимо як σ^k_i.

Розраховуємо значення щільності *D(i)* для особини *i*:

$$D(i) = \frac{1}{\sigma_i^k + 2}, k = \sqrt{(N + N_A)}$$

1.4. Остаточне значення функції пристосованості *F(i)* для особини *i* визначається, як

$$F(i) = R(i) + D(i)$$

Крок 2. Модернізація архіву. Створити проміжний архів A_t*= P_t.

а) скопіювати особини, чиї вектори рішень недоміновані щодо P_t в A_t*.

б) видалити тих індивідів з A_t*, чиї відповідні вектори рішень слабко доміновані щодо A_t*.

в) зменшити число індивідів, що зберігаються в архіві, і помістити результуючу зменшену множину особин в A_{t+1}. Для зменшення кількості индивидів в архіві використовується наступна процедура:

2.1. Всі недоміновані особини (значення функцій пристосованості яких менше 1) з архіву і популяції копіюються в архів наступної ітерації:

$$A_{t+1} = \{i | i \in P_t \cup A_t \wedge F(i) < 1\}, i = \overline{1, N}$$

2.2. Якщо розмір утвореного архіву відповідає бажаному ($|A_{t+1}|=N_A$), то кластеризація завершується. Інакше можливі дві ситуації:

2.3а. Розмір утвореного архіву менше бажаного ($|A_{t+1}|<N_A$).

У цьому випадку особини з об’єднаної множини $P_t + A_t$ сортуються у порядку зростання значень їх функцій пристосованості і перші N_A - $|A_{t+1}|$, особин з відсортованої множини копіюються в архів A_{t+1}

2.3б. Розмір утвореного архіву більше бажаного($|A_{t+1}|>N_A$).

У цьому випадку починається процедура зменшення розміру архіву, яка ітеративно видаляє особини з A_{t+1}, поки $|A_{t+1}|=N_A$:

1) Створюємо порожній список *D*

2) Для кожної особини *i* обчислюється її декартова відстань *d(i,j)* до всіх решти особин *j*. Найменша з отриманих відстаней додається до списку

$$D = D + d_i, d_i = min\ d(i, j)$$

3) Значення отриманого списку *D* сортуються за зростанням. Особина, що відповідає найменшому значення зі списку *D* видаляється з архіву.

4) Даний процес повторюється, поки архів не досягне бажаного розміру N_A.

Крок 3. Ранжування: Особини в популяції сортуються за значеннями функцій пристосованості за спаданнями (від найкращої особини до найгіршої)

Крок 4. Групування: особини діляться на групи, кожна з яких складається з двох особин. Ці дві особини обираються з початку списку відсортованих особин.

Крок 5. Схрещування і мутація: в кожній зі сформованих груп відбувається схрещування (кросовер) та мутація.

Схрещування. Для будь-яких двох індивідів необхідність виконання операції схрещування визначається випадковим чином:

5.1) Випадковим чином генерується число p_c* з проміжку [0,1]. Отримане число порівнюється із заданою ймовірністю схрещування.

Якщо $p_c^* > p_c$, то схрещування не відбувається і батьківські особини залишаються без змін. У протилежному випадку — відбувається процес схрещування (див. пункт 5.2).
5.2) В даному алгоритмі використовується варіант одноточкового кроссоверу — два вибраних "батьки" перерізуються у випадково обраній точці, після чого їх хромосоми обмінюються своїми фрагментами.
Мутація. Необхідність виконання мутації визначається аналогічно до подібної для операції схрещування:
5.3) Випадковим чином генерується число p_m^* з проміжку [0,1]. Отримане число порівнюється із заданою ймовірністю мутації.
Якщо $p_m^* > p_m$, то мутація не відбувається і батьківські особини залишаються без змін. У протилежному випадку — відбувається процес мутації (див. пункт 5.4).
5.4) Мутація пов'язана з випадковою зміною одного чи декількох генів у хромосомі. З двох батьківських особин формується дві дочірні особини. Батьківські особини видаляються з групи.

Крок 6. Усі дочірні особини об'єднуються у одну групу, яка стає новою популяцією P_t.

Крок 7. Закінчення: Покласти $P_{t+1} = A_t$ і $t = t + 1$. Якщо $t \geq T$ або виконується якийсь інший критерій зупинки, тоді A_t - є шукана множина розв'язків, інакше перейти на крок 1.

Крок 8. "Лікування точок"

Еволюційні алгоритми непогано справляються з безумовними задачами багатокритеріальної оптимізації, але при розв'язанні задач з обмеженнями отримані рішення не завжди задовільні:

- вони можуть не містити точку умовного максимуму
- результуючі точки можуть бути розкидані в пошуковому просторі
- частина із знайдених рішень може лежати за межами дозволеної області

Для усунення виявлених недоліків генетичних алгоритмів пропонується проводити «лікування» (уточнення) недомінованих точок, отриманих після зупинки генетичного алгоритму.
8.1. Вибрати деякий початковий допустимий розв'язок (хромосому) X
8.2. Сформувати множину хромосом $Y = \{y_1, ..., y_k\}$, де хромосома y_i отримана внаслідок мутації i-го біту хромосоми X, $i = \overline{1,k}$, k — кількість бітів в хромосомі.
8.3. Обчислити значення функцій пристосованості для всіх хромосом з множини Y, використовуючи процедуру описану на **кроці 2** та обрати серед них хромосому Y^* з найкращим значенням функції пристосованості
8.4. Якщо $F(Y^*) < F(X)$ (для задачі максимізації), то розв'язок X є найкращим в даному околі. Переходимо до кроку 8.1 та обираємо наступну точку для покращення, доки не будуть "вилікувані" усі розв'язки.
8.5. Інакше, покласти $X = Y^*$. Перейти на крок 8.2.

Крок 9. Ліквідація згустків точок

9.1. Для кожного індивіда $i \in A$ утворюємо окремий кластер C_i.
9.2. Якщо $|C| \leq N_A$, перейти на Крок 9.5, інакше перейти на Крок 9.3.
9.3. Обчислити відстань між усіма можливими парами кластерів. Віддаленість d_c двох кластерів C_i і $C_j \in C$ визначається як середня відстань між парами індивідів, що належать цим кластерам:

$$d_c = \frac{1}{|C_i| \cdot |C_j|} \cdot \sum_{c_i \in C_i, c_j \in C_j} \|c_i - c_j\|$$

де оператор ||.||відображає евклідову відстань (в просторі цілей) між двома особинами c_i та c_j, оператор |.| визначає кількість елементів у множині.
9.4. Визначити два кластери C_i і $C_j \in C$ з мінімальним відстанню d_c. Об'єднати ці кластери в один більший за розміром кластер C_{ij} та перейти на крок 9.2.

$$C_{ij} = C_i \cup C_j$$

9.5**.** Для кожного кластера вибрати репрезентативного індивіда, а всіх інших індивідів з нього видалити. (Таким чином, репрезентативний індивід — це центроїд, точка з мінімальною середньою відстанню до всіх інших точок кластера.) Визначити зменшений архів шляхом об'єднання репрезентативних індивідів всіх кластерів.

Отримана в результаті роботи запропонованого алгоритму апроксимація множини Парето є репрезентативною - точки рівномірно розподілені, згущення відсутні, домінованих точок немає.

Висновки

Запропонований алгоритм, який дозволяє отримати набір глибоких нейронних мереж оптимальної конфігурації з точки зору точності роботи та складності, додатково враховуючи можливі обмеження щодо результуючої топології.

Література

1. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm. In Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2001.
2. Watanabe S., Hiroyasu T., Miki M. "NCGA: Neighborhood Cultivation Genetic Algorithm for Multi-Objective Optimization Problems", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002), p. 458–465, 2002.
3. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. In Evolutionary Computation, Vol. 8(2), pp. 173–195, 2000.

RESUME

D.Y. Koval, V.M. Sineglazov, O.I. Chumachenko

Synthesis of optimal structure of deep neural network

This paper is devoted to the design of a new algorithm for structure-parametrical synthesis of Deep Belief networks (DBN's). DBN's are generative models that contain many layers of hidden variables. The main building block of a DBN is a bipartite undirected graphical model called a restricted Boltzmann machine (RBM). A good estimate of the partition function would be extremely helpful for model selection and for controlling model complexity, which are important for making RBM's generalize well. It is important to decrease the complexity of DBNs under accuracy preservation.

The proposed algorithm is based on evolutional approach for the solution of conditional multicriterion problem. Each chromosome will correspond to a certain neural network architecture. The number of neurons in the first and last layer can be determined by the task. In this case, only the rest of the components of each vector are randomly generated.

It is considered the procedure of transformation this problem into unconditional multicriterion problem. Described algorithm among modified genetic operators as crossover and mutation, also uses Pareto dominance concept for evaluation of fitness value for each individual (and corresponding neural network structure as result) and also includes additional steps ("chromosome healing" and clustering, which guarantee, that set of received solutions will include optimal structure and increase its representativeness – number of competing possible solutions.

The use of the algorithm permits to determine the quantity of layers, neurons in every layer of based network and correspondingly the quantity of RBMs which include the DBN.

Надійшла до редакції 18.10.2016

